Lecture 8: October 17, 2001

- Review: Rabin’s OTS
- Public key cryptography
  - RSA
  - Etc.

- HW0 due today
- HW1 due next Monday (October 22)
- Midterm: Wednesday (October 24)
Rabin’s One-Time Signatures

$K_i \sim \text{random secret keys } (0 < i < 2n)$

Signature:  $S_i = E(K_i, H(MSG))$

Public keys:  $P_i = E(K_i, i)$

$|i| = l \quad E\text{-block}\text{-size}$

$\Omega = \{R_i \mid 0 < i < 2n\}$

$\Sigma = \{S_i \mid 0 < i < 2n\}$,  

$P = \{P_i \mid 0 < i < 2n\}$,  

Pick $n$ random numbers:  

$R_i, \quad 0 < i < 2n$

$n$ – security parameter

Inefficient

On-line

$D(K_{R_i}, S_{R_i}) = H(MSG)$

$D(K_{R_i}, P_{R_i}) = i$

10/19/01  

Gene Tsudik, ICS 268 Winter 2001
Rabin’s One-Time Signatures (contd)

**DISPUTES**

\[
\{ K_i \mid 0 < i < 2n \}
\]

Has P

\[
MSG, \; \Sigma
\]

1) \( D(K_i, P_i) \) ?
   if not, Bob wins

2) \( E(K_i, H(MSG)) \) ?
   for at most \( n \) \( S_i \)
   Alice wins, else Bob wins

Note: Alice, Bob communicate with Court via authentic channels
Bob Cheats:

If Bob attempts to forge Alice’s signature on a new message MSG’, B either:

- needs to determine at least one more key $k_{n+1}$
- or
- determine MSG’ such that $H(MSG’) = H(MSG)$.

This should be infeasible if the symmetric-key algorithm $E()$ and hash function $H()$ are chosen appropriately.

Alice Cheats:

If Alice attempts to create a signature which it can later repudiate, she must ensure that precisely $n$ $S_i$-s contain $H(MSG)$ while the other $n$ contain $H(MSG’)$ where $MSG <> MSG’$. She can then hope that B chooses exactly these $n$ values in the verification procedure, the probability of which is only:

$$\frac{2n!}{n!(2n-n)!}$$
Diagram illustrating symmetric encryption:

- **Alice**: Sender
- **Bob**: Receiver

The process is as follows:

1. **Encryption**: $M \rightarrow \text{ciphertext} \leftarrow Enc_k^M$
2. **Decryption**: $\text{ciphertext} \rightarrow M \leftarrow Dec_k^M$

Gene Tsudik, ICS 268 Winter 2001
Symmetric authentication

Alice

k

M → M, Auth_k(M)

Auth_k

Bob

k

M, Auth_k(M) → "OK"

Verify_k

Gene Tsudik, ICS 268 Winter 2001
Public Key (asymmetric) encryption

Alice

Bob

encryption

M $\rightarrow$ ciphertext

Dec$^{\text{privkey}}$

ciphertext $\rightarrow$ M

decryption

Gene Tsudik, ICS 268 Winter 2001
Public Key (asymmetric) authentication

M → M, Sign_{privkey}(M) → M, Sign_{privkey}(M) → “OK”
Digital Signatures

• A public key technique to authenticate information in a way that uniquely binds the signer to the signature

• Can be used to provide non-repudiation and may be legally binding

Recipient knows:
1. that the message is that of the supposed sender
2. can prove (a) to a third party
Why Public Key is important

• Reduces from $N^2$ to $N$ the number of keys needed in a distributed setting (network)

• Less reliance on a “trusted center” for system availability and secrecy (e.g., electronic cash)

• Non-repudiation (Notarization)
Virtual Private Networks (VPN)
Let $n = pq$ where $p \neq q$ - (large) primes

$e, d \in \mathbb{Z}_n$ and $e = d^{-1}$ and $ed \equiv 1 \mod \Phi(n)$

note that: $\Phi(n) = (p - 1)(q - 1)$

Secrets: $p, q, d$

Publics: $n, e$

Encryption: message = $m$

$E(x) = y = m^e \mod n$

Decryption: ciphertext = $y$

$D(y) = x' = y^d$
Primality Testing

- Needed to generate $p, q$
- Fact: primes are not that rare! $N/(\ln N)$
- E.g., for 512-bit modulus, $P(p\text{-prime})=1/177$
- So, all we do is generate 1000 or so random integers... and test them...
- Solovay/Strassen and Miller/Rabin: yes-biased Monte-Carlo composite testing algorithms.
- Both run in $O((\log n)^3)$ time; must be re-run $x$ times until $[P\_error(n)]^x$ -- sufficiently small!
Primality Testing

- Monte Carlo algorithm
  - Yields yes/no answer
  - One is always correct
  - The other may be incorrect with prob.™
  - Yes-biased and No-biased

- Las Vegas algorithm
  - Yields a correct answer
  - May not give any answer (with prob.™)
Other RSA odds and ends

- If Eve discovers decryption exponent (d), then she can use it to factor n... => e,d must be changed together with n!

- RSA decision problems are as hard as decryption!
  - Parity: Is last bit of cleartext 1 or 0?
  - Half: Is cleartext > n/2?
  - Let’s try half(y)=0, half(E(2x))=0 => ?????

  Bit security <=> cleartext security! (see pp. 144-145)

- RSA is malleable! Eve can tinker with ciphertext...

- For true security need randomized (probabilistic) encryption with built-in integrity/redundancy

- Ask me later if interested...
Using half() to decrypt RSA

- Suppose a “black box” machine that correctly answers queries of the form:
- Assume 2-bit message: X in (00, 01, 10, 11)
- Y = RSA(X)
- Half(Y)=1 => X₁=1
- Half(Y)=0 => X₁=0
- Half(Y*RSA(2)) mod n = 1 => => X₀=1
- Half(Y*RSA(2)) mod n = 0 => => X₀=0
Rabin PK cryptosystem (78-79)

\[ p \leftrightarrow q - \text{large primes} \]
\[ p, q \equiv 3 \mod 4 \]
\[ n = pq \]
\[ B \in [0, n[ \]
\[ P, C = Z_n \]

publics - \( n, B \)
secrets - \( p, q \)

Encryption : \( E_k(x) = y = x(x + B) \mod n \)
\[ x^2 + Bx - y = 0 \]

Decryption : \( D_k(y) = \sqrt{B^2 / 4 + y - B / 2} \)

Let \( C = B^2 / 4 + y \)
and \( X = x + B/2 \)
then :
\[ X^2 \equiv C \mod n \]

equivalent by :
\[ X^2 \equiv C \mod p \]
\[ X^2 \equiv C \mod q \]
Why?

\[ \text{since } p, q \equiv 3 \mod 4 \]
\[ X = \pm C^{(p+1)/4} \mod p \hspace{5mm} \text{and} \hspace{5mm} X = \pm C^{(q+1)/4} \mod q \]
then use CRT
(4times)
to find \( X \)

Using Euler’s criterion
Euler’s Criterion:

If $p$ is an odd prime, $X$ is a quadratic residue (mod $p$) iff:

$$X^{(p-1)/2} \equiv 1 \mod p$$